Terrain Height Evidence Sharing for Collaborative Autonomous Rotorcraft Operation

Eric N. Johnson,
eric.johnson@ae.gatech.edu
Lockheed Martin Associate
Professor of Avionics Integration

John G. Mooney,
john.g.mooney@gatech.edu
Graduate Research Assistant

Georgia Institute of Technology, Atlanta, Georgia

Matthew White,
matthew.white@sikorsky.com,
Engineer, Handling Qualities and Control Laws

Jonathan Hartman
jonathan.hartman@sikorsky.com
Engineer, Advanced Concepts

and Vineet Sahasrabudhe
vsahasrabudhe@sikorsky.com
Manager, Handling Qualities and Control Laws
Sikorsky Aircraft Corporation, Stratford, Connecticut

This paper describes recent results from a partnership between the Sikorsky Aircraft Corporation and the Georgia Institute of Technology to develop, improve, and flight test a sensor, guidance, navigation, control, and real-time information sharing system to support collaborative autonomy and high performance nap-of-the-Earth helicopter flight. The emphasis here is on smart and selective sharing of terrain data which (1) minimizes the bandwidth consumed by obstacle/terrain-information-sharing between aircraft, (2) assigns an appropriate level of confidence to the data received from other heterogeneous aircraft, (3) is robust to sensor error and failures, and (4) is robust to entry and exit of vehicles from the network. Results from simulation and flight testing are provided.

Introduction

Unmanned aerial vehicles (UAVs) and optionally piloted aircraft are expected to play an increasingly important role in both civil and military applications. Civil applications include, among others, law enforcement, traffic and weather reporting, cargo transport, agriculture, and communications relay. Many of these functions are difficult to fully accomplish using a single aircraft; rather, a dispersed network of aircraft collaborating autonomously would provide more complete sensor coverage while being more robust to environmental disturbances and aircraft failure.

Multi-robot coordination and information sharing has been studied by many researchers, primarily in the last 20 years. These efforts have investigated multiple paradigms of control and coordination. Arkin[1], Balch[2], and Parker[3], for example, have focused on reactive behavior-based control and interaction. This approach assumes minimal or no direct communication between robots, often relying upon the robot's ability to observe the behavior of other robots to coordinate efforts. Other efforts have maintained decentralized control, but allowed robots to explicitly share state information [4]. Yet others use a fully-centralized approach, treating the system as a single "meta-robot" with a very high-dimension configuration space [5][6]. Coordination of robotic aircraft has been studied extensively as well, though usually in the context of collision avoidance or formation control [7][8][9].

Under this effort, a largely de-centralized approach to aircraft coordination has been studied. The work here presents a novel solution that allows autonomous aircraft to share the necessary terrain information with other aircraft at a frequency that maintains a "nearly" common operating picture of the environment without excessively consuming bandwidth. Additionally, this solution allows the aircraft to effectively incorporate or not incorporate the information received from another aircraft as appropriate.

Presented at The AHS International Specialists' Meeting on Unmanned Rotorcraft and Network Centric Operations,

The remainder of this paper is organized as follows. First, the methodology by which the aircraft select and compact their terrain information updates is explained, as is the method used to incorporate updates from other aircraft. Then, the obstacle avoidance methodology which makes use of the shared terrain data is described. Third, a description of the aircraft utilized for simulation and flight test evaluation is included. Finally, simulation and flight test results are included.

General Approach

Prior work used three-dimensional evidence grids to maintain a model of the local terrain. This approach was used successfully for obstacle avoidance and had previously been used by Scherer, et al. in a similar manner [10]. A brief summary of the approach is included here.

An evidence grid represents a space of interest as a three dimensional array, using it to store a measure of likelihood of a particular point in space being occupied. This likelihood is updated when the sensor receives a measurement, according to rules specific to the sensor's characteristics. A particularly useful version of this evidence grid uses log-odds as the measure of likelihood or 'belief':

$$b(o) = \ln\left(\frac{P(o|M)}{1 - P(o|M)}\right)$$

If subsequent measurements are assumed to be independent, then this belief function can be updated by adding (or subtracting) the log-odds of occupancy according to the sensor model and new measurements.

$$b_{new}(o) = b_{old}(o) + \ln\left(\frac{P(M_{new}|o)}{1 - P(M_{new}|o)}\right)$$

The last term on the right hand side of (2) is a characteristic of specific sensor chosen. The lack of multiplies/divides allows this information to be stored as an integer type. In particular, a single byte contains enough range to represent near certainty of either non-occupancy or occupancy and everything in between.

b(o)	**P(o|M)**
-127	6.99×10^{-56} %
0	50%
127	99.99 . . . 93 %

The evidence grid has two primary desirable properties which led us to seek to modify it rather than select a new scheme to map the terrain. First, it is robust to sensor error, in that false positive readings can quickly be erased from the map by additional true readings in the same area. Second, it is already structured to update the map based on probabilistic evidence. Finally, previous research under this effort has validated the practical utility of this method.

Unfortunately, even with each array element occupying only a single byte, the size of a reasonably large three dimensional grid with appropriate resolution can consume significant network bandwidth. For example, a relatively small "local area" 128 x 128 x 64 grid consumes megabyte of memory. Refining the resolution by a factor of two over the same space would result in an eight megabyte map. In this research, we anticipate operating aircraft in relatively close quarters, with map data sharing at 1 Hz or faster. Accordingly, a more compact version of this mapping method is needed to enable rapid data sharing.

Here, a few simplifying assumptions about the terrain allows the use of a pair of two dimensional grids. One 2D grid is used to store a number representing terrain height, and the other represents the confidence level associated with that terrain height. This configuration might be suitably called "2-½" dimensional and effectively means we do not care about free space under a known object. That is, we will not intentionally fly under anything. Under these assumptions, the evidence update rules are modified slightly. If new data indicates higher terrain than in the current map, the map is automatically updated to reflect this terrain height. If new data indicates the terrain is actually lower than what is mapped, the evidence grid is decremented. If the evidence of terrain height drops below 50%, then the terrain height at that spot is decremented. Figures 1 and 2 summarize this update methodology in graphic form.

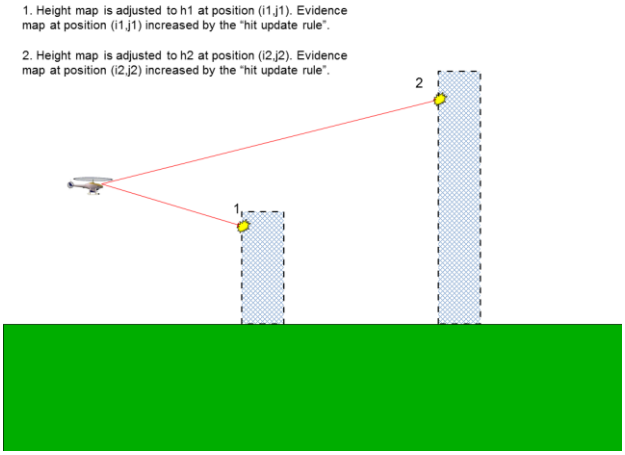


Figure 1. Illustration of Terrain Height Map Update Rules -- Initial

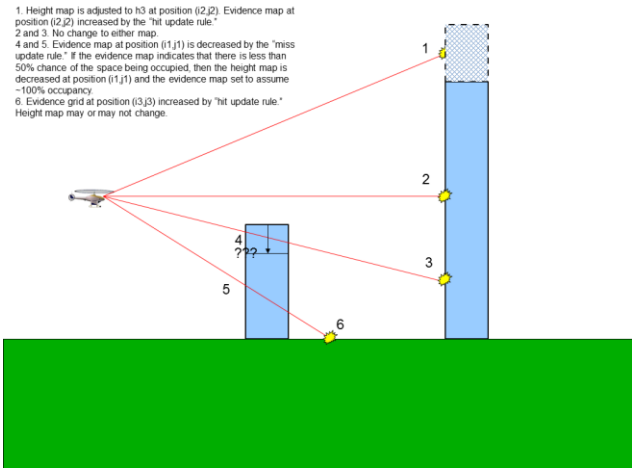


Figure 2. Illustration of Terrain Height Map Update Rules -- Subsequent

Data Sharing Algorithm

In order to share this information, each aircraft must consider: which information do I send to the others? Which information do I accept from others? If the two conflict, then how to I resolve the difference?

Here we assume that each aircraft trusts its own sensors over those of others. This is a reasonable assumption for shared terrain data, as terrain sensors on the aircraft are subject to fewer errors (mainly platform location and time delay) and are likely from the most relevant perspective. Second, we assume that recent observations are more trustworthy than older observations.

In light of these assumptions, the aircraft keep a third array in parallel with the terrain height map and evidence map: a map recording the time of last observation. When an update is due to be sent, the aircraft finds all the points that have been observed and changed terrain height since the last update.

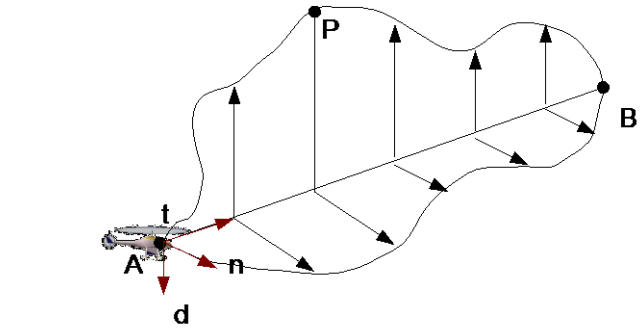
When the aircraft receives an update from the others in the network, it accepts the other aircraft data unless the data conflicts with own-ship data *and* the own-ship data is younger than a specified age.

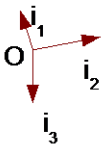
Finally, the aircraft share their state data in order to facilitate collision avoidance. This data is a trivial amount of information compared to the terrain data itself. This data-sharing framework is used in the work here for terrain mapping and obstacle avoidance. However, it can easily be extended to be used for the general case of sharing any probabilistic measurements of the environment, such as object tracking, radiation monitoring, or wind mapping.

Obstacle Avoidance

This data is used with an obstacle avoidance methodology to utilize the terrain sharing. The algorithm has been used in previous work and is summarized below.

This method uses a simple set of six motion primitives to optimize lateral and vertical cross-tracking error from a straight-line path, while treating the obstacle field as a constraint set (figure 3). The use of motion primitives for autonomous systems, including aircraft, is well known [19] [20] [21]. This particular method selects an optimal combination of a relatively small set of sinusoidal primitives. These primitives reduce the problem to six dimensions, the space of which contains a set of intuitive and "good enough" solutions. These primitives also have properties that make a number of later computations simpler. The solver searches in seven arbitrary directions to find basic feasible solutions, then searches from those points by gradient descent to find an optimum.





$$\vec{r}_p(s) = \vec{r}_A + s\,\vec{t} - l(s)\,\vec{n} - v(s)\,\vec{d}$$

$$D = \|\vec{r}_B - \vec{r}_A\|,\ \vec{t} = \frac{\vec{r}_B - \vec{r}_A}{D},\ \vec{n} = \frac{\tilde{i}_3 \vec{t}}{\|\tilde{i}_3 \vec{t}\|},\ \vec{d} = \tilde{t}\,\vec{n}$$

$$l(s) = l_1 \sin\left(\frac{s\pi}{D}\right) + l_2 \sin\left(2\frac{s\pi}{D}\right) + l_3 \sin\left(3\frac{s\pi}{D}\right)$$

$$v(s) = v_1 \sin\left(\frac{s\pi}{D}\right) + v_2 \sin\left(2\frac{s\pi}{D}\right) + v_3 \sin\left(3\frac{s\pi}{D}\right)$$

Figure 3. Illustration of Motion Primitives

To prevent collisions, the aircraft make an estimate of each other's future position, and treat a ball around that position as an obstacle. This path plan is updated every time new obstacle information is received, either

through the aircraft's own sensors or when receiving data from another aircraft.

Experimental Procedures

Test Aircraft

A Yamaha RMAX based research UAV, Figure 4, was utilized for the simulation and flight test activities under this effort. The system consists of four major elements: the basic Yamaha RMAX airframe, a modular avionics system, baseline software, and a set of simulation tools.

Figure 4. Yamaha RMAX based research UAV utilized for this effort, 10.2 ft main rotor

The hardware components that make up the baseline flight avionics include general purpose processing capabilities and sensing. The research avionics configuration includes:

- 2 Embedded PCs
- Inertial Sciences ISIS-IMU Inertial Measurement Unit
- NovAtel OEM-4, differential GPS
- Sick LD-MRS laser scanner, Figure 5
- Custom made ultra-sonic sonar altimeter
- Honeywell HMR-2300, 3-Axis magnetometer
- Actuator control interface
- Vehicle telemetry (RPM, Voltage, Remote Pilot Inputs, low fuel warning)
- 11 Mbps Ethernet data link and an Ethernet switch
- FreeWave 900MHz serial data link

Figure 5. Sick LD-MRS Laser scanner mounted under the nose of the aircraft, able to see down and forward (sensor rotated 90 degrees in roll, 40 degrees nose down pitch)

The baseline navigation system running on the primary flight computer is a 17 state extended Kalman filter. The states include: vehicle position, velocity, attitude (quaternion), accelerometer biases, gyro biases, and terrain height error. The system is all-attitude capable and updates at 100 Hz [11]. The baseline flight controller is an adaptive neural network trajectory following controller with 18 neural network inputs, 5 hidden layer neurons, and 7 outputs for each of the 7 degrees of freedom [12]. These 7 degrees of freedom include the usual 6 rigid-body degrees of freedom plus a degree of freedom for rotor RPM. The baseline flight controller and navigation system, which coupled with the simple baseline trajectory generator, is capable of automatic takeoff, landing, hover, forward flight up to the maximum attainable by the helicopter and aggressive maneuvering.

Test Scenarios

A set of four different scenarios were developed to exercise and demonstrate the aspects of this information sharing scheme. They represent progressively more complex scenarios.

In the first scenario, the two aircraft fly independently of one another in two non-overlapping flight areas. The objective of this test is to check that the aircraft are both sending and receiving accurate terrain updates without the complication of overlapping data. The expected outcome of this scenario is that the aircraft have essentially identical terrain maps.

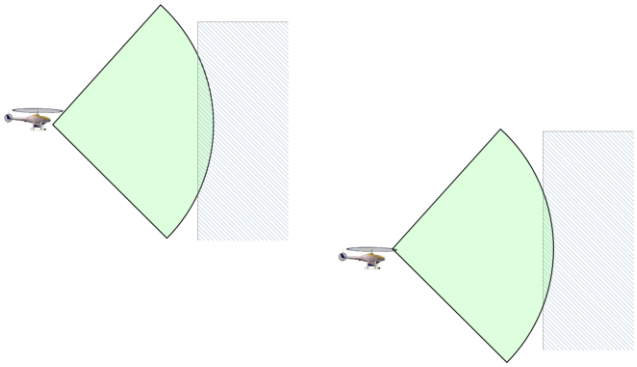


Figure 6. Illustration of first test scenario, sharing of independent observations

The second scenario is designed to test the ability of one aircraft to be robust to erroneous data sent from the other aircraft. The first aircraft captures false readings and transmits them to the other aircraft; the second aircraft then scans the same area, correcting its own map and simultaneously sending the updates to the first. The first aircraft then updates its own map. The expected outcome of this test is the false readings are wiped out from both aircraft's map and the end maps are again essentially identical.

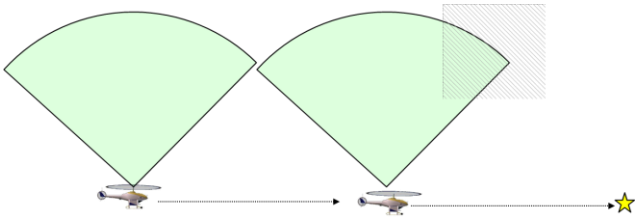


Figure 7. Illustration of second test scenario, second aircraft correcting false data from first

The third scenario is designed to test the sharing of aircraft state data, as well as the anti-collision logic in the guidance algorithm. The aircraft are given intersecting flight paths in an obstacle-free flight area and commanded to proceed to their next waypoint. One or both of the aircraft should recognize an imminent collision and adjust its flight path accordingly.

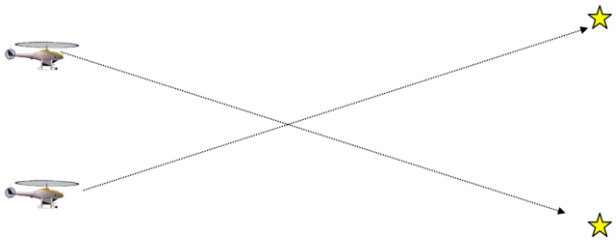


Figure 8. Illustration of third test scenario, verifying anti-collision logic

The final scenario culminates the series of tests by having one aircraft avoiding obstacles solely based upon the other's measurements. The expected behavior of the second aircraft is to successfully avoid the obstacle while staying primarily within space that the other aircraft has observed.

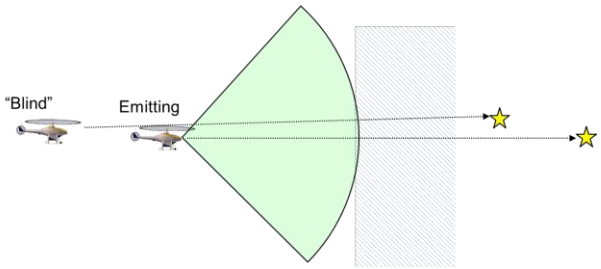


Figure 9. Illustration of fourth test scenario, a "blind" aircraft relying solely on another aircraft's observations

Results

Simulation

Flight control software was developed utilizing the existing Georgia Tech UAV Simulation Tool (GUST), which is a set of C/C++ software that supports pure software, hardware-in-the-loop, and research flight test operations [13]. GUST includes models of the sensors, aircraft, and aircraft interfaces – down to the level of binary serial data (i.e., packets). It enables injection of model error and environmental disturbances. It includes a flexible scene generation capability and reconfigurable data communication routines, enabling a large number of possible hardware-in-the-loop simulation configurations.

Each of the four scenarios were tested in this simulation environment prior to flight test. The simulated environment is a model of McKenna MOUT site, Fort Benning, Georgia, in order match the future flight test environment. Both aircraft were simulated on the same computer in order to isolate the data-sharing algorithm from hardware or network problems. Later tests on multiple machines revealed similar results.

The terrain maps of the two aircraft after scenario 1, shown in figures 10 and 11, demonstrate success of the terrain sharing architecture. The maps are virtually identical (figure 12), with the only difference being due to a slight datum offset between the two aircraft.

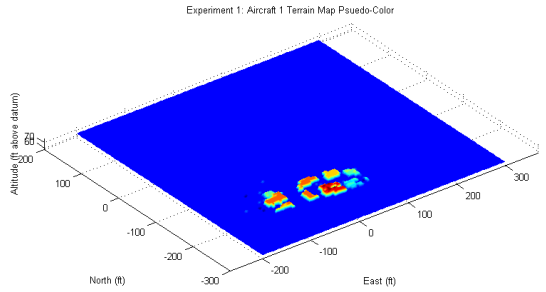


Figure 10. Terrain map from aircraft 1, scenario 1

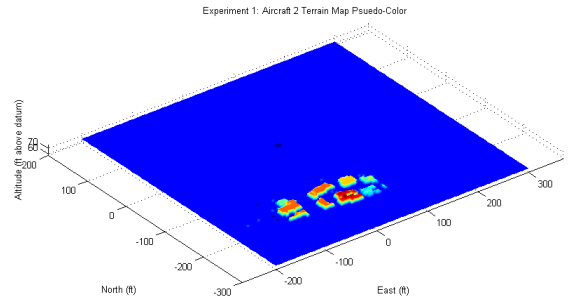


Figure 11. Terrain map from aircraft 2, scenario 1

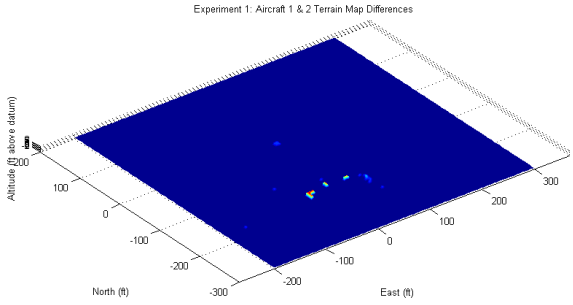


Figure 12. Difference of terrain maps, scenario 1

The second scenario was successful as well. The first aircraft observed a simulated building and added it to the map (figure 13). The second aircraft observed the same area but without the building, correcting the erroneous information it received. Notably, the rate of erasure of the other aircraft's false data was highly dependent upon the speed of the second aircraft. If it is too fast, there aren't enough observations to wipe out the erroneous ones.

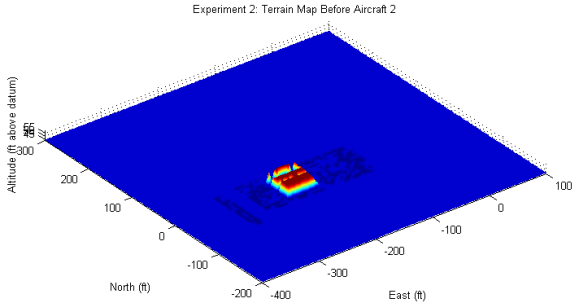


Figure 13. Terrain map after "faulty" observations, scenario 2

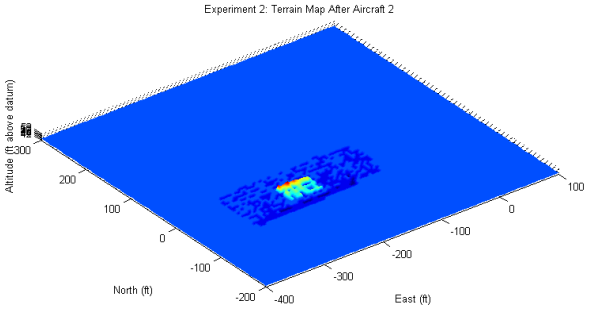


Figure 14. Terrain map with corrected observations, scenario 2

The third scenario verified the anti-collision logic. Shown in figure 15, the aircraft with the yellow flight path can be seen to make a slight maneuver to the right, then to the left, to avoid the aircraft with the yellow path.

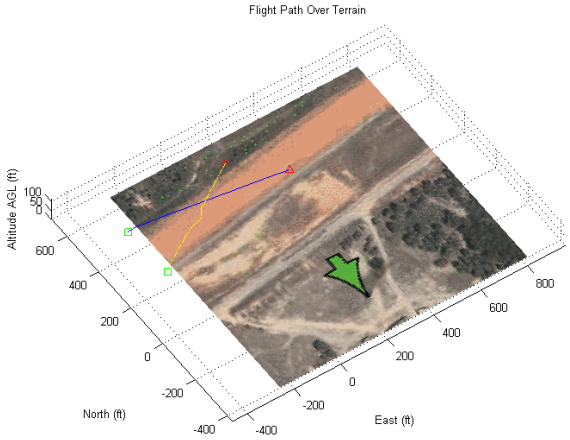


Figure 15. Flight paths in the third test scenario, checking anti-collision logic

The fourth scenario was successful as well. The lead aircraft began by observing a simulated tree, then selecting a path over and to the north of it. The second aircraft, being blind, selected a path to the right as well, favoring space that it knew had been observed.

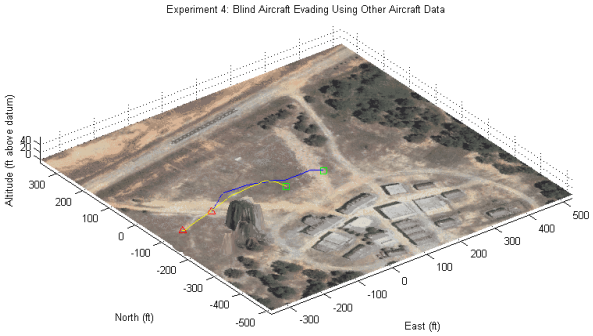


Figure 16. Flight paths of the two aircraft in the fourth scenario

Flight Testing

Table 2: Weather Conditions at Fort Benning, GA During Flight Tests (MFBGG1) [10]

Parameter	27 July 2012
Mean Temperature (deg F)	85
High Temperature (deg F)	90
Mean Wind Speed (mph)	5 WSW
Precipitation (in)	None

The four scenarios were executed in flight test in late July 2012 at McKenna MOUT site, Fort Benning, Georgia using the physical GTMax aircraft and another, identical RMax simulated at the ground station.

The flight test results largely matched results found in simulation, with a few notable exceptions. First, the data from the real laser proved to be much "messier" than the simulated laser, mainly because the real terrain and buildings aren't perfect planes and polygons like the simulator. This is born out in the figures 17 through 21 below, where "smears" in the maps arise from the assumed ground plane being at a slightly different height than the actual ground. Another interesting effect is the combining of maps constructed from real and simulated laser data. Note in figures 17 and 18, the terrain map on the right side of the map—from the simulated aircraft—is much cleaner and the buildings more distinct. Finally, the avoiding aircraft in scenario 3 took an up and over avoidance approach, probably due to a slight difference in timing from the simulation (figure 22).

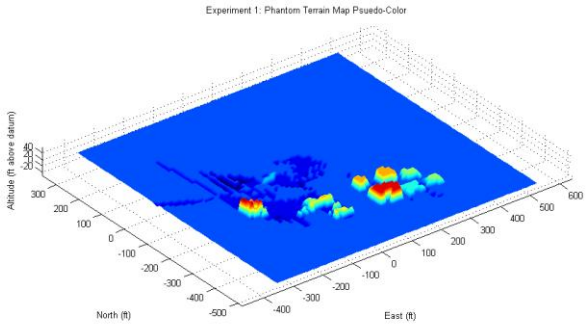


Figure 17. Terrain map from aircraft 1, scenario 1

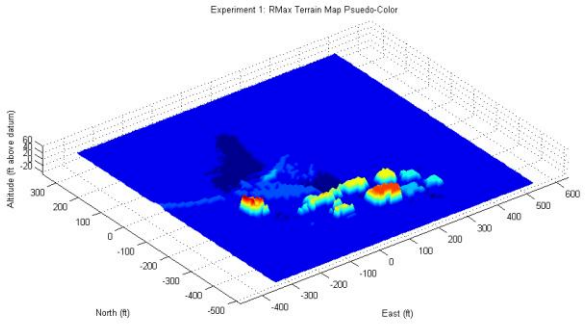


Figure 18. Terrain map from aircraft 2, scenario 1

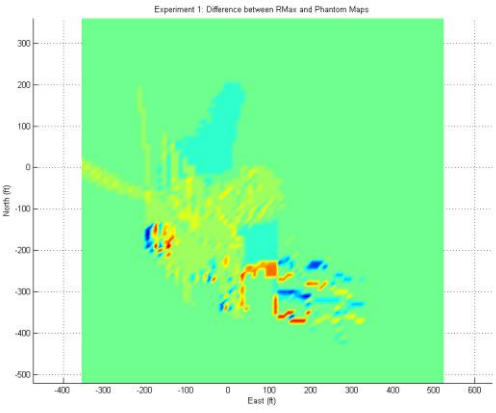


Figure 19. Difference of terrain maps, scenario 1

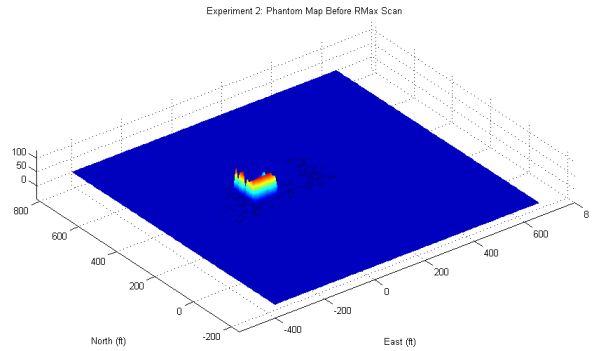


Figure 20. Terrain map after "faulty" observations, scenario 2

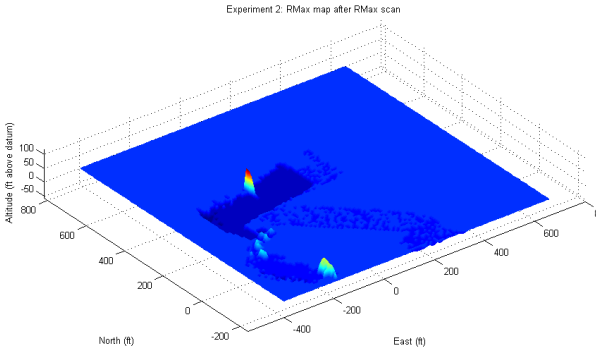


Figure 21. Terrain map with corrected observations, scenario 2

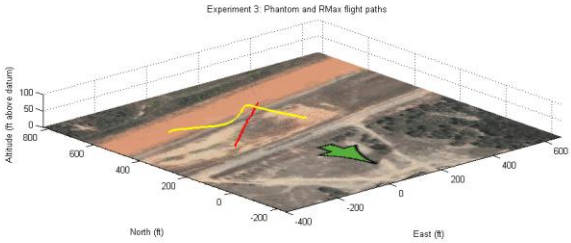


Figure 22. Flight paths in the third test scenario, checking anti-collision logic

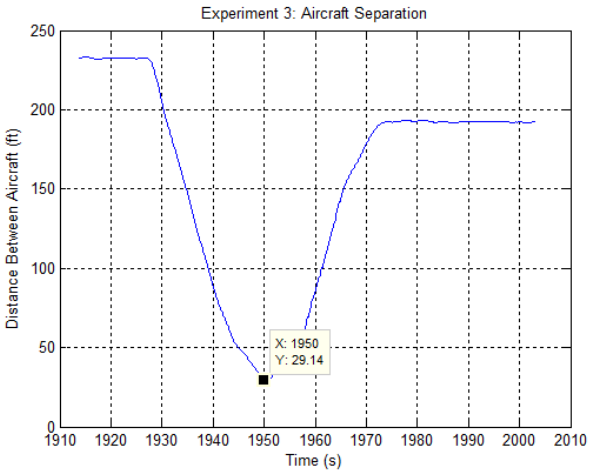


Figure 23. Absolute aircraft separation, third scenario

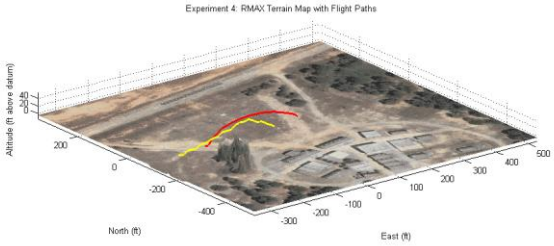


Figure 24. Flight paths in the fourth test scenario

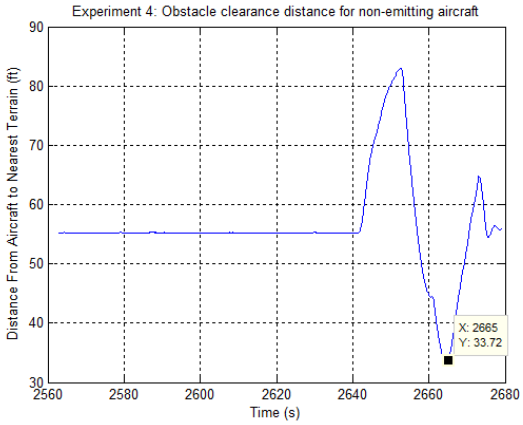


Figure 25. Separation from terrain, non-emitting aircraft, fourth scenario

Conclusions

This paper describes the effort to test a multi-vehicle information sharing system. A simple, but robust method of terrain data sharing was demonstrated, along with a very basic form of collaborative obstacle and collision avoidance. The algorithms worked as designed in both simulation and flight test. Future work in this area includes the flight test of two real aircraft, as well as expansion of the evidence map method to other applications.

Acknowledgements

The authors would like to acknowledge support from Sikorsky Aircraft Corporation and the Georgia Institute of Technology, and the help of other contributors including: Jeong Hur, Dmitry Bershadsky, Russ Halstead, Claus Christmann, Gerardo Delatorre, Daniel Magree, and Fritz Langford.

References

[1] Balch, T., & Arkin, R. (1998). Behavior-based formation control for multirobot teams. *Robotics and Automation, IEEE Transactions, XX*, 1–15.

[2] Balch, T. (2000). Social Potentials for Scalable Multi-Robot Formations. *International Conference on Robotics and Automation*.

[3] Parker, L. (1994). *Heterogeneous multi-robot cooperation*. Retrieved from http://oai.dtic.mil/oai/oai?verb=getRecord&metadataPrefix=html&identifier=ADA279839

[4] Bruce, J., & Veloso, M. (2002). Real-time randomized path planning for robot navigation. *Robots and Systems, 2002. IEEE/RSJ*.

[5] van Den Berg, Jur, et al. "Centralized path planning for multiple robots: Optimal decoupling into sequential plans." *Robotics: Science and systems V* 2 (2009).

[6] Luna, Ryan, and Kostas E. Bekris. "Efficient and complete centralized multi-robot path planning." *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*. IEEE, 2011.

[7] Godbole, D.N.; , "Control and coordination in uninhabited combat air vehicles," *American Control Conference, 1999. Proceedings of the 1999* , vol.2, no., pp.1487-1490 vol.2, 2-4 Jun 1999

[8] Ribichini, Gabriele, and Emilio Frazzoli. "Efficient coordination of multiple-aircraft systems." *Decision and Control, 2003. Proceedings. 42nd IEEE Conference on*. Vol. 1. IEEE, 2003.

[9] Anderson, Brian, et al. "UAV formation control: Theory and application." *Recent Advances in Learning and Control* (2008): 15-33.

[10] S. Scherer, S. Singh, L. Chamberlain, M. Elgersma, "Flying Low and Fast Among Obstacles: Methodology and Experiments," The International Journal of Robotics Research, Vol. 27, No. 5, 549-574 (2008).

[11] Dittrich, J.S. and Johnson, E.N., "Multi-Sensor Navigation System for an Autonomous Helicopter," Proceedings of the 21st Digital Avionics Systems Conference, October 2002.

[12] Johnson, E.N. and Kannan, S.K., "Adaptive Trajectory Control for Autonomous Helicopters," AIAA Journal of Guidance, Control, and Dynamics, Vol. 28, No. 3, pp. 524-538, May/June 2005.

[13] Johnson, E.N. and Schrage, D.P., "System Integration and Operation of a Research Unmanned Aerial Vehicle," AIAA Journal of Aerospace Computing, Information, and Communication, Vol. 1, No. 1, pp. 5-18, January 2004.